FieldServer Technologies

-

A Sierra Monitor Company

Driver Manual
(Supplement to the FieldServer Instruction Manual)

FS-8700-103 ATMI-TGM Serial Driver

APPLICABILITY & EFFECTIVITY

Effective for all systems manufactured after May 1, 2001

Driver Version:	**1.00**
Document Revision:	**2**

TABLE OF CONTENTS

1. ATMI TGM Serial Driver Description

The ATMI-TGM driver allows the FieldServer to transfer data to and from devices over RS-232 using the ATMI-TGM printer port protocol.

This driver connects to an ATMI TGM device using a serial connection. One device may be connected per FieldServer port. (This limitation exists because the protocol is node-less; that is, messages do not contain information about the node and thus messages from different TGM devices could not be distinguished if they were connected on the same port.)

The driver reports Gas and sensor readings and Gas alarm / warning status. The driver reports TGM module malfunctions.

The driver is a client only driver.

When reporting port and sensor data, the driver reports the gas name by setting a gas name index value. The index value is obtained from a list of gas names / index values hard coded into the driver. The list may be extended and/or modified using the configuration by using appropriate entries in the configuration CSV file. The driver reports a second gas name index value based on reading the TGM calibration report. When the report is read each gas is allocated an identifier value based on the gas's position in the configuration report.

The driver does not provide emulation of a TGM device and thus may not be used as a server.

It is not possible to use this driver to write to the TGM device.

Max Nodes Supported

FieldServer Mode	**Nodes**	**Comments**
CLIENT	1	ONLY 1 SERVER NODE ALLOWED PER PORT.
SERVER	1	Server capability not provided.

2. Driver Scope of Supply

2.1. Supplied by FieldServer Technologies for this driver

FieldServer Technologies PART #	Description
FS-8917-12	Connector, 9-pin female: connects to DTE, DSR/DTR loop
FS-8700-103	Driver Manual.

2.2. Provided by the Supplier of 3rd Party Equipment

2.2.1. Required 3rd Party Hardware

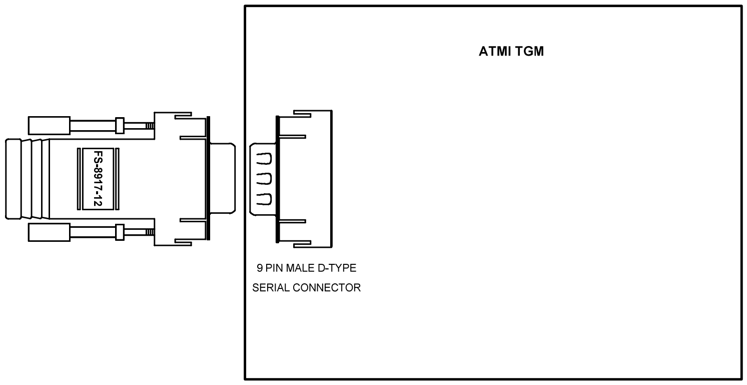
Part #	Description
No specific Hardware requirements other than TGM unit.	

2.2.2. Required 3rd Party Configuration

The FieldServer's serial port must be configured to be used as a 'Console Port' on the TGM device. The 'Console Port' is configurable. At least one of the two serial ports on the TGM device must be configured to enable the 'Console Port'. This is not always the case when the TGM is shipped. Notes in Appendix D provide additional information.

3. Hardware Connections

The FieldServer is connected to the TGM device's printer port as shown in connection drawing. Configure the TGM device according to manufacturer's instructions



WIRE LIST

FS-X40 FUNCTION	FROM	TO	COLOR
RX	RJ45-01	DB9F-03	GREY
GND	RJ45-04	DB9F-05	GREEN
TX	RJ45-08	DB9F-02	BLUE
DTR		DB9F-04	BLACK
DSR		DB9F-06	YELLOW
CTS		DB9F-08	BROWN

FieldServer Technologies (408)-262-2299

ATMI TGM
CONNECTION DIAGRAM

BASE NAME: FILE NAME: T28700-103.vsd	DATE: 04/30/04 BY: MN

3.1. Hardware Connection Tips / Hints

The following notes are provided from a TGM manual:

The TGM code constantly checks the status of the DSR input of its serial ports connected to the DTR output of the remote serial ports. If the TGM DSR is NOT ASSERTED (FALSE), then the TGM software will suspend serial data output out of TGM Tx, and start a timeout timer (the default timeout time = 30 seconds as set in CONFIG.DAT). During this timeout, if the DSR is ASSERTED (TRUE), then the TGM software will begin transmitting data. If the DSR hold exceeds the (configurable) 30 second timeout, the TGM will report a malfunction:

```
    MALFUN      PRINTER OFF LINE                        hh:mm dd mmm yy
    MALFUN      REMOTE OFF LINE                         hh:mm dd mmm yy
```

depending upon which port has timed out. Restoration of the DSR status as ASSERTED or TRUE will clear the above malfunction(s). The above malfunctions will occur if the active TGM serial ports (printer or remote) are not connected upon bootup with the proper null modem cables to active serial ports with DTR (connected to theTGM DSR) ASSERTED/TRUE. The TGM can have the hardware control defeated by jumpering CTS to DTR to DSR on the TGM side of the connector.

The driver does not provide the hardware handshaking required and assumes the hardware control has been defeated with jumpers.

4. Configuring the FieldServer as a ATMI TGM Serial Client

For a detailed discussion on FieldServer configuration, please refer to the FieldServer Configuration Manual. The information that follows describes how to expand upon the factory defaults provided in the configuration files included with the FieldServer (See ".csv" sample files provided with the FieldServer).

This section documents and describes the parameters necessary for configuring the FieldServer to communicate with a TGM device.

4.1. Data Arrays/Descriptors

The configuration file tells the FieldServer about its interfaces, and the routing of data required. In order to enable the FieldServer for ATMI TGM Serial communications, the driver independent FieldServer buffers need to be declared in the "Data Arrays" section, the destination device addresses need to be declared in the "Client Side Nodes" section, and the data required from the servers needs to be mapped in the "Client Side Map Descriptors" section. Details on how to do this can be found below.

Note that in the tables, * indicates an optional parameter, with the bold legal value being the default.

Section Title
Data_Arrays

Column Title	Function	Legal Values
Data_Array_Name	Provide name for Data Array	Up to 15 alphanumeric characters
Data_Array_Format	Provide data format. Each Data Array can only take on one format.	Float, Bit, UInt16, SInt16, Packed_Bit, Byte, Packed_Byte, Swapped_Byte
Data_Array_Length	Number of Data Objects. Must be larger than the data storage area required by the Map Descriptors for the data being placed in this array.	1-10,000

Example

```
//    Data Arrays
Data_Arrays
Data_Array_Name,          Data_Format,          Data_Array_Length
DA_AI_01,                 UInt16,               200
DA_AO_01,                 UInt16,               200
DA_DI_01,                 Bit,                  200
DA_DO_01,                 Bit,                  200
```

4.2. Client Side Connection Descriptions

Section Title
Connections

Column Title	Function	Legal Values
Port	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[1]
Protocol	Specify protocol used	TGM-Serial, ATMI-TGM-Serial
Baud*	Specify baud rate	110 – 19200, standard baud rates only (Vendor limitation)
Parity*	Specify parity	**None** (Vendor limitation)
Data_Bits*	Specify data bits	**8** (Vendor limitation)
Stop_Bits*	Specify stop bits	**1**
Handshaking*	Specify hardware handshaking	**None**
Poll _Delay*	Time between internal polls	0-32000 seconds, **1 second**

Example

```
//        Client Side Connections

Connections
Port,          Protocol,          Baud,     Parity,       Handshaking,        Poll_Delay
P8,            TGM-Serial,        9600,     None,          None,              0.100s
```

4.3. Client Side Node Descriptors

Section Title
Nodes

Column Title	Function	Legal Values
Node_Name	Provide name for node	Up to 32 alphanumeric characters
Node_ID	This parameter is not required for the TGM driver. However, if you intend to use the Node_Status function with a Data Array then the Node_ID must be specified.	1-255
Protocol	Specify protocol used	TGM-Serial, ATMI-TGM-Serial
Connection	Specify which port the device is connected to the FieldServer	P1-P8, R1-R2[1]

Example

```
//    Client Side Nodes

Nodes
Node_Name,                 Node_ID,               Protocol,               Connection
PLC 1,                     1,                     TGM-Serial,             P8
```

[1] Not all ports shown are necessarily supported by the hardware. Consult the appropriate Instruction manual for details of the ports available on specific hardware.

4.4. Client Side Map Descriptors

4.4.1. FieldServer Related Map Descriptor Parameters

Column Title	Function	Legal Values
Map_Descriptor_Name	Name of this Map Descriptor	Up to 32 alphanumeric characters
Data_Array_Name	Name of Data Array where data is to be stored in the FieldServer	One of the Data Array names from “Data Array” section above
Data_Array_Offset	Starting location in Data Array	0 to maximum specified in “Data Array” section above
Function	Function of Client Map Descriptor. The driver cannot write to the TGM device.	RDBC, ARS

4.4.2. Driver Related Map Descriptor Parameters

Column Title	Function	Legal Values
Node_Name	Name of Node to fetch data from	One of the node names specified in “Client Node Descriptor” above
Data_Type	This commonly used driver parameter is not used by the driver.	
Length	Length of Map Descriptor The parameter is not used by the driver but is used by the FieldServer Kernel to reserve space in the Data Array. Set the Length according to the function.	QIR Report: Length = 100 CALR Report: Variable. Depends on number of gases reported in CALR report. Start with a length of 100 and increase if necessary. QLA Report: If the config has no sensors and port x is the max port number, then make the array (x+1) * 10 elements long If the config has any sensors and sensor x is the max sensor number then make the array 200 + (x+1) * 10 elements long
Address	This commonly used parameter is ignored by the driver.	

4.4.3. Timing Parameters

Column Title	Function	Legal Values
Scan_Interval	Rate at which data is polled For CALR report use a long interval (600s) as the TGM configuration seldom changes. Set the Scan_Interval for the QLA and QIR reports to zero to have the driver poll for this data as often as possible.	≥0.001s

4.4.4. Map Descriptor Example #1 – CALR Report.

To be useful to the upstream device the driver reports gas concentration values and gas names from the QLA report. In reporting the gas names it is most useful when a driver stores an index value which is used to imply the gas names. The reason this is most useful is that it is typically easier for automation devices to work with numbers and not text.

When the driver reads the Calibration report, it does two things.
1) It stores a string of gas names vs. index values in the Data Array. This string can be viewed using the Ruinet Utility and may be of help in interpreting the gas name index values stored by the QLA report.
2) The data is stored internally by the driver and when a QLA report is parsed the driver looks the gas names up in the CALR report data and uses the lookup to determine what gas name index value to store.

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name,  Data_Array_Name,  Data_Array_Offset,  Function,  Node_Name,  Address  Length,  Scan_Interval,  TGM_Funtion
RD_CALR,              DA_CALR,          0                   RDBC,      Node_A,     0,       100,     600s,           CALR
```

Although not mandatory it is recommended that each MD is given a unique name.

A list of gas names vs. index values is stored in the Data Array specified here.

In this particular example, the list is stored as an ASCII string and hence it makes most sense to ensure that the Data Array's format is BYTE so that it can be viewed as a string when using Ruinet.

Data storage starts at this location in the Data Array.

Tells the driver to repeat this read task continuously. The polling interval is set by the Scan_Interval parameter.

For the CALR report is may also make sense to use the 'ARS' function which gets the driver to read this data once each time the driver restarts.

The node name connects the Map Descriptor to a Node definition which in turn is connected to a port. That is how the driver knows which port to use to poll for this data.

Reserve enough space in the Data Array to store the gas/index string.

No need to read this report too often.

Tells the driver to read the Calibration report.

4.4.5. Map Descriptor Example #2 – QLA Report.

In this example, a Map Descriptor is created to read the Latest Area Report (QLA). This report contains gas concentration and status for each port and sensor. Ten Data array elements are required for each port / sensor that is reported. The poistion in the array is dependent on the port/sensor number. Appendix B provides additional information. By specifying the extra arrays (DA_Bit_Name and DA_Byte_Name) the driver stores additional information.

```
//    Client Side Map descriptors

Map Descriptors
Map_Descriptor_Name,  Data_Array_Name,  Data_Array_Offset,  Function,  Node_Name  Address  Length  Scan_Interval  DA_Byte_Name   DA_Bit_Name   TGM_Funtion
RD_QLA,               DA_CALR,          0,                  RDBC,      Node_A,    0,       300,    1.0s           DA_GAS_UNITS   DA_QLA_GAS    QLA
```

Gas concentration value, status, engineering units index values, gas name index values are stored in this Data Array

This parameter may be omitted. If specified then the Data Array will be used to store the gas concentration value engineering units in ASCII. Storage position is dependent on the port/ sensor number.

This parameter may be omitted. If specified then the Data Array will be used to store the gas name in ASCII. Storage position is dependent on the port/ sensor number.

Tells the driver to read the QLA report.

4.5. Map Descriptor Example #3 – QIR Report – Malfunctions

In this example the Map Descriptor is configured to read the incident report. The driver parses the report looking for malfunctions and reports that show a malfunction has been cleared. Other incidents / events are ignored except for events #19 and #37 which cause all malfunctions to be cleared. The Driver updates the array setting values to 1 (malfunction) or zero when a malfunction has been cleared. The position indicates the malfunction number. The driver has a list of malfunctions and index numbers stored internally. Refer to Appendix A.2 for this list. The list can be modified in the configuration CSV file.

CAUTION: When the QIR report is polled it responds with the latest incidents only. This means that care should be taken to synchronize the FieldServer and the TGM. Read more about this in Appendix B.8

```
//    Client Side Map Descriptors

Map Descriptors
Map_Descriptor_Name,    Data_Array_Name,    Data_Array_Offset,    Function,    Node_Name    Address    Length    Scan_Interval    TGM_Funtion
RD_QIR,                 DA_MALFUNC,         0                     RDBC,        Node_A,      0,         100,      2.0s             QIR
```

Driver sets the Data Array elements to 1 when there is malfunction and sets the value to zero when the malfunction is cleared.

Position in the array indicates the malfunction number.

Tells the driver to read the incident report.

Appendix A. TGM Tables

Appendix A.1. Hard Coded Gas Name Table

This table is used by the driver when storing data from the QLA report. The driver looks the reported gas name up in this table. If a match is found then the driver stores the corresponding index value in the Data Array. If no match is found the the driver stores a value of –1.

The table may be extended/modified using the configuration file. Refer to Appendix B.4

Gas Name	Index Value
Error / Unknown	-1
NONE	1
ASH3	3
B2H6	6
CH4	9
CH4AC	12
CL2	15
DET	18
F123	21
GEH4	24
H2	27
H2AC	30
H2S	33
H2SE	36
HCL	39
HF	42
N2O	45
NF3	48
NH3	51
O2	54
O3	57
PH3	60
POCL	63
POCL3	66
SIHX	69
SIHx	70
TMBP	75
WF6	78

Appendix A.2. Malfunction Table

When an event is reported in the Incident Report (QIR) then the driver compares the event description to text in the table below. If the text matches an an entry in the 'Malfunction' column (and is preceeded by the string MALFUNC) then the Data Array element at the specified offset is set to 1. If the text matches the 'Return to Normal' text then the malfunction is considered as cleared and the Data Array element is set to zero.

The Data Array elements are all set to zero when
Event 19 reports " ALARM RESET " *or* Event 37 reports "ALL MALFUNCTIONS CLEAR"

The Malunction text strings generated by the TGM unit are not the same for all units. They can be customized in theTGM's config.dat file. For this reason, this driver allows the strings in the table below to be customized too – see Appendix B.5. In addition to allowing this table to be customised in the configuration file, the driver can use a config.dat file to directly customise the table – see Appendix B.7

Offset	Malfunction	Return to Normal
20	BASELINE OUT OF RANGE	BASELINE BACK TO NORMAL
22	IGNITER SWITCH IS ON	IGNITER SWITCH IS OFF
24	FLAME IGNITION DISABLED	FLAME IGNITION ENABLED
26	LOW VACUUM	VACUUM BACK TO NORMAL
28	FLAMEOUT RESTART	FLAMEOUT BACK TO NORMAL
30	LINE LEAK TEST MALFUN	LINE LEAK BACK TO NORMAL
32	POWER FAILURE	POWER RESTORED
38	FAILED TO LOAD PROGRAM	
39	DPM TIMEOUT	DPM BACK TO NORMAL
41	RELAY FILE ERROR	OPERATIONS CHECK DONE
45	CHECK I/O AND FUSES	I/O AND FUSES OK NOW
47	TGM IN DEBUG MODE	
48	DISK NEARLY FULL	ADEQUATE DISK STORAGE NOW
50	FLAMEOUT CONDITION	RETURN FROM FLAMEOUT
52	FAILED FLAME TEST	FLAME TEST OK NOW
54	FAILED RESPONSE TEST	RESPONSE TEST OK NOW
56	FAILED BLOCK LEAK CHECK	BLOCK LEAK CHECK OK NOW
58	ANALOG SUBSYSTEM FAILURE	ANALOG SUBSYSTEM NORMAL
60	ANALOG CHANNEL FAILURE	ANALOG CHANNEL NORMAL
62	PRINTER OFF LINE	PRINTER BACK TO NORMAL
64	REMOTE OFF LINE	REMOTE BACK TO NORMAL
66	DISK READ/WRITE ERR	DISK BACK TO NORMAL
68	LAN READ/WRITE ERROR	LAN BACK TO NORMAL
70	DISK FULL - DATA LOST	DISK NO LONGER FULL
75	HIGH SAMPLE VARIANCE	SAMPLE VARIANCE OK NOW
80	COUS SENSOR TIMEOUT	COUS TIMEOUT CORRECTED
82	COUS GETTER INOPERATIVE	COUS GETTER OK NOW
84	ACOUSTIC SENSOR FAILURE	ACOUSTIC SENSOR RECOVERD
86	GETTER TEMPERATURE LOW	GETTER TEMPERATUR OK NOW
88	LOW SAMPLE FLOW	SAMPLE FLOW OK NOW
90	TGM HYDROGEN LEAK	HYDROGEN LEAK REPAIRED

Appendix A.3. Gas Concentration Engineering Units

This table is used by the driver when storing data from the QLA report. The driver looks the reported engineering units up in this table. If an matching entry is found then the driver stores the index value. If a matching entry cannot be found then the driver stores a value of –1. (Note that when –1 is stored in a BYTE or UINT16 formatted Data Aray it will appear as a positive number such as 255 (Byte Array) or 65535 (uint16 Array).

The table may be extended/modified using the configuration file. Refer to Appendix B.1

Engineering Units	Index Value
Error / Unknown	-1
ARU	1
PPM	2
PPB	3
LEL	4
%	5
PCT	6

Appendix B. Advanced Topics

Appendix B.1. How Malfunctions are determined and stored

- The notes below describe how the driver parses incident reports obtained by the QIR poll.
- The driver starts at the end of the report and works its way to the top, line by line.
- Each line is inspected to see if it contains either of the following strings
  "ALARM RESET"
  "ALL MALFUNCTIONS CLEAR"
- If either is reported then the driver sets all the array elements to zero except those corresponding to lines of the report appearing after (and hence later in time) one of the above lines that indicates that malfunctions be set, in which case those particular malfunctions will be reported as 1's in the Data Array.
- When either one of these two events is found the driver stops parsing the report as all prior lines predate the clear event.
- Each line is inspected for the string "MALFUN". If it is found then 'Event' field in the report is inspected and the description is extracted and compared to the malfunction list provided in Appendix A.2. If a match is found the driver uses the offset value and sets the corresponding element in the Data Array to 1.
- If the string "MAFUN" is not found then the description found in the 'Event' field of the report is used to look up a 'Return to Normal' event in the table provided in Appendix A.2. If a match is found then the corresponding offset in the Data Array is set to zero.
- If the description is not recognized then it it is ignored.

Example

Only the Malfunction 'COUS SENSOR TIMEOUT' would have its Data Array element set to 1. All others will be set to zero. The reason is that all events above the 'ALARM RESET' line are considered to be prior to the reset and are thus cleared when the driver sets all the Data Array elements to zero. The timeout event occurs later and hence its malfunction is set.

```
MALFUN  POWER FAILURE                     00:00 00     00
        POWER RESTORED                    10:29 16 FEB 04
MALFUN  TGM IN DEBUG MODE                 10:29 16 FEB 04
MALFUN  FAILED FLAME TEST                 10:29 16 FEB 04
        ALARM RESET                       10:29 16 FEB 04
MALFUN  COUS SENSOR TIMEOUT               10:29 16 FEB 04
```

When the driver uses the lookup table to see if it recognizes the event description then it compares the strings looking for an exact match. If no exact match is made then the driver uses a so-called 'soundex' function to see if the strings are very similar, if they are then they are considered to be matched. If a similar match is made the driver reports this in the error log.

```
16 T02> TGM:#21 FYI. Malfunction. Dont recognize <POWER FAILED>
17 T02>                    But sounds like <POWER FAILURE>.
```

Appendix B.2. How Gas Data is Stored

- The QLA reports provides Gas Concentration values for each configured port/sensor connected to the TGM.
- The driver stores the data using the "Data_Array_Name" and "Data_Array_Offset" parameters specified on the Map Descriptor. Two additional Data Array's can be specified to tell the driver where to store the gas names and egineering units.
- The position at which data is stored in the array is deprendent on the TGM port/sensor number. Use the following formula to determine the position.
  For Port x : Relative Offset = Port_Number * 10
  For Sensor x : Relative Offset = 200 + (Sensor_Number * 10)
- Thus enough room is provided to store data from 20 ports and provided the Array is long enough, there is no limit to the maxium sensor number.
- The offset's described here are relative to the offset provided in the Map Descriptor, which should be added to the offsets specified here

Relative Offset	Contents
In the Data Array specified by the "Data_Array_Name"	
0	Port Number
1	Set to 'P' for a port. Set to 'S' for a sensor
2	Gas concentration Value. Can be scaled by applying Map Desc scaling using the parameters Data_Array_Low_Scale, Data_Array_High_Scale, Node_Low_Scale, Node_High_Scale
3	Gas Conc. Engineering Units as an index value. Stores –1 if units are not recognized. Index value is based on lookup table - Appendix A.3
4	Gas Status. 0=Normal; 1=Warn; 2=Alarm; 4=Error
5	Gas Name index (based on CALR report) A list of gas names and index values is built when the CALR report is parsed. It is not possible to provide a list of index values vs gas names in advance because this is dependent on the TGM configuration. To facilitate debugging your application the driver stores the list it builds in a Data Array which can be read by an upstream device or which can be seen by uisng the Ruinet Utility. If the gas name is not recognized then the driver stores a value of –1.
6	Gas Name index based on looking up the gas name in the table provided in Appendix A.1 If the gas name is not recognized by the lookup then a value of –1 is stored.
In the Data Array specified by the "DA_Bit_Name"	
0	Gas Name is stored character for character to a max of 9 characters. The name of the Map Desc parameter is confusing. The Data Array specified with this parameter can have any format - 'BYTE' format is reccomended.
In the Data Array specified by the "DA_Byte_Name"	
0	Gas Concentration Engineering Units are stored character for character to a max of 9 characters. The name of the Map Desc parameter is confusing. The Data Array specified with this parameter can have any format and in fact the 'BYTE' format is reccomended.

Appendix B.3. Adding/Modifying Gas Concentration Value Engineering Units

The configuration CSV file can be used to modify the table of engineering units that is hard coded into the driver. This table is presented in Appendix A.3

Add a section to the configuration file which is similar to the following example.

In the example below the Units specified is 'PPM', an entry from the existing list. When the driver finds the lines below, it will update the index value to 20.

```
// Change index value of existing unit to 20.

Driver_Table
TGM_Units_Name,                 TGM_Units_Index_Value,       Protocol
PPM,                            20,                          TGM
```

In the example below the units specified are 'gpl' and they have been assigned an index value of 7.

```
// Add a new engineering unit

Driver_Table
TGM_Units_Name,                 TGM_Units_Index_Value,       Protocol
gpl                             7,                           TGM
```

Appendix B.4. Adding / Modifying Hard Coded Gas Names

When a gas concentration is reported, the driver reports the gas name by looking up the gas name in a list of hard coded gas names. When a match is found the driver stores the associated index value. The table may be modified/extended in the configuration CSV file. The table of names and index values is provided in Appendix A.1

Add a section to the configuration file which is similar to the text found in the examples below.

In the example below the gas name specified is 'H2', an entry from the existing list. When the driver finds the lines below, it will update the index value to 20.

```
// Change index value of existing gas to 20.

Driver_Table
TGM_Gas_Name,                   TGM_Gas_Index_Value,         Protocol
H2,                             20,                          TGM
```

In the example below the gas name specified is 'C2H5OH' and it has been assigned an index value of 7. It will be added to the table.

```
// Add a new gas name

Driver_Table
TGM_Gas_Name,                 TGM_Gas_Index_Value,       Protocol
C2H5OH,                       7,                         TGM
```

Appendix B.5. Adding / Modifying Malfunction Strings

When an event is reported, the driver compares the "MAFLUN" events to those in the table provided in Appendix A.2. The table can be modified or extended.

Add a section to the configuration file which is similar to the text found in the examples below.

In the example below the gas name specified is 'DISK NEARLY FULL', an entry from the existing list. When the driver finds the lines below, it will update the index value to 20. Ensure the index value is less than 100.

```
// Change index value of existing malfunction.

Driver_Table
TGM_Malfunction_Name,         TGM_Malfunction_Index,     Protocol
DISK NEARLY FULL,             20,                        TGM
```

In the example below the gas name specified is 'DISK IS FULL' and it has been assigned an index value of 20. It will be added to the table.

```
// Add a new malfunction

Driver_Table
TGM_Malfunction_Name,         TGM_Malfunction_Index,     Protocol
DISK IS FULL,                 20,                        TGM
```

Appendix B.6. Adding / Modifying 'Return to Normal' Strings

When an event is reported, the driver compares the description to those in "Return to Normal" column of the table provided in Appendix A.2. The table can be modified or extended.

Add a section to the configuration file which is similar to the text found in the examples below.

The index value allocated should match the malfunction index for the driver to clear the malfunction when the return to normal event is found.

In the example below the gas name specified is 'DISK NO LONGER FULL, an entry from the existing list. When the driver finds the lines below, it will update the index value to 20. Ensure the index value is less than 100.

```
// Change index value of existing Return to Normal Event.

Driver_Table
TGM_Event_Normal_Name,      TGM_Event_Normal_Index,     Protocol
DISK NO LONGER FULL,        20,                         TGM
```

In the example below the gas name specified is 'DISK NOT FULL' and it has been assigned an index value of 20. It will be added to the table.

```
// Add a new Return to Normal Event

Driver_Table
TGM_Event_Normal_Name,       TGM_Event_Normal_Index,     Protocol
DISK NOT FULL,               20,                         TGM
```

Appendix B.7. Using the TGM's Config.dat file to customize malfunctions

Copy the TGM's config.dat file to the FieldServer as a file called atmitgm.ini. When the driver starts it looks for this file and updates the malfunction table based on the malfunction strings found inside the config file. The Ruinet utility is required

Ruinet –i1.2.3.4 –u0 –lc:\…\config.dat –fatmitgm.ini

Where
    1.2.3.4 is the IP address of the FieldServer
    c:\….\config.dat is the path of the file to be downloaded from your C driver.

To stop the driver using the ini file remove it from the FieldServer by deleting it.

Ruinet –i1.2.3.4 –zatmitgm.ini

When the driver inspects the configuration file it searches for the string "malfunStat" and uses the table that follows to update the malfunction table. Note that the config file only contains malfunction strings and does not provide the 'Return to Normal' strings. Thus, if the config file contains a malfunction for which the driver does not have a matching return to normal string then you will need to add the matching ‘Return to Normal’ string. This process is described in Appendix B.6

Appendix B.8. Synchronizing Malfunctions

THE Incident Report (QIR) only reports events which occurred since the previous poll.

The following procedure is recommended for synchronizing the FieldServer and the TGM

1. Turn off the FieldServer. Clear all events on the TGM and then restart the FieldServer.

   Or

2. With the FieldServer online, cause the TGM to send an RESET ALARMS message to the FieldServer.

Appendix C. Driver Error Messages

Message	Notes and Corrective Action~
TGM:#1 Err. Diagnostic #1.	If this message is printed an internal diagnostic has been generated. Contact FieldServer tech support for help.
TGM:#2 Err. TGM_Function not specified.	Each Map Descriptor must have the parameter called "TGM_Function" set. See section 4.4.2[2]
TGM:#3 FYI. Use an Array called <%s> to expose diagnostics.	This message may be ignored. For more information read Appendix C.1
TGM: #4c Err. Bad format in latest area report. Discarding data.	4a: The text string "CUR=" could not be found in the QLA report. 4b: The text string "GAS=" could not be found in the QLA report. 4c: The text string "STATUS=" could not be found in the QLA report. If the error is printed rarely and data is being stored correctly then the error probably arises from a corrupted message. If the error occurs frequently then it is possible that the report format has changed and the driver firmware may require an update. To facilitate the evaluation of the problem, take a serial communications log, ensuring at least one of the errors occurs during the log, and then contact Tech Support.
TGM: #5 FYI. atmitgm.ini being used for malfunctions	This message is printed to alert you to the fact that the malfunction list is being based on the contents of a TGM config file. Refer to Appendix B.7. If you expect the driver to be using this list, ignore the message.
TGM: #6 FYI. You could have used a TGM config file to define malfunctions. Working with hard coded list.	This message is printed to alert you to the possibility that the TGM config file can be used with this driver. Refer to Appendix B.7.
TGM: #7 Err. Driver cannot write. MD=%s	The Map Descriptor function has been set so that the driver will attempt to write to the TGM. Writing is not possible. Change the function to a read.[2]
TGM: #8 Err. Array=%s too short. Act/Rqd=%d/%d	This error is printed when the driver attempts to store data from the QLA report. Modify the length of the Data Array as discussed in section 4.1.[2]
TGM: #9 FYI. Duplicate Gas Name=<%s>. Value has been updated from=%d to=%d	This message may be safely ignored. It is printed to alert you to the fact that the driver has updated the Gas Name table so that default values have been modified.

[2] Modify the configuration CSV file. Download the modified file to the FieldServer and Reset the FieldServer for the changes to take effect.

Message	**Notes and Corrective Action˜**
TGM: #10 Err. No space. Reject Gas Name=<%s> Index Value=%d	An attempt has been made to extend the gas name table. There is insufficient space to add more names. Remove this entry from the configuration CSV file[3].
TGM: #11 FYI. User added Gas Name=<%s> Index Value=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the Gas Name table by adding a new gas name and index value. Ensure that index values are unique.
TGM: #12 FYI. Duplicate Gas Units=<%s>. Value has been updated from=%d to=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the Gas Engineering Unit table so that default values have been modified.
TGM: #13 Err. No space. Reject Gas Units=<%s> Index Value=%d	An attempt has been made to extend the Fas Engineering Units table. There is insufficient space to add more names. Remove this entry from the configuration CSV file.[3]
TGM: #14 FYI. User added Gas Units=<%s> Index Value=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the Gas Engineering Units Table by adding a new gas name and index value. Ensure that index values are unique.
TGM: #15 FYI. Duplicate Malfunction=<%s>. Index Val has been updated from=%d to=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the Malfunction Table so that default values have been modified. Ensure that the index value is less than 100.
TGM: #16 Err. No space. Reject Malfunction=<%s> Index Val=%d	An attempt has been made to extend the Malfunction Table. There is insufficient space to add more names. Remove this entry from the configuration CSV file.[3]
TGM: #17 FYI. User added Malfunction=<%s> Index Val=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the Malfunction Table by adding a new gas name and index value. Ensure that index values are unique and that the index value is less than 100.
TGM: #18 FYI. Duplicate Normal Event=<%s>. Index Val has been updated from=%d to=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the 'Return to Normal' Table so that default values have been modified. Ensure that the index value is less than 100.
TGM: #19 Err. No space. Reject Normal Event=<%s> Index Val=%d	An attempt has been made to extend the 'Return to Normal' Table. There is insufficient space to add more names. Remove this entry from the configuration CSV file.[3]
TGM: #20 FYI. User added Normal Event=<%s> Index Val=%d	This message may be ignored. It is printed to alert you to the fact that the driver has updated the 'Return to Normal' Table by adding a new gas name and index value. Ensure that index values are unique and that the index value is less than 100.

[3] Modify the configuration CSV file. Download the modified file to the FieldServer and Reset the FieldServer for the changes to take effect.

Message	**Notes and Corrective Action˜**
TGM: #21 FYI. Malfunction. Don’t recognize <%s> But sounds like <%s>.	The driver has found a malfunction but could not match it exactly to a malfunction in the Malfunction Table. The driver used a soundex function which looks for matches based on a ‘sounds like’ evaluation. As the match is not exact the driver provides the alert. To suppress this message in the future, note the message and add it to the Malfunction Table. Notes in Appendix B.5 provide information on how to do this.
TGM: #22 FYI. Return to Normal. Don’t recognize <%s>But sounds like <%s>.	The driver has found an event description that is being evaluated as a potential ‘Return to Normal’ message, but could not match it exactly to a ‘Return to Normal’ in the Malfunction Table. The driver used a soundex function which looks fro matches based on a ‘sounds like’ evaluation. As the match is not exact the driver provides the alert. To suppress this message in the future, note the message and add it to the ‘Return to Normal' Table. Notes in Appendix B.6 provide information on how to do this.
TGM: #23 FYI. Duplicate Malfunction Number=<%d>.Text has been updated from=<%s> to=<%s>	If these messages are produced then the driver processes the TGM config file. Read Appendix B.7 for more information.
TGM: #24 Err. No space. Reject Malfunction=<%s> Index Val=%d	
TGM: #25 FYI. User added Malfunction=<%s> Index Val=%d	
TGM: #26a Err. Malfunction=%s Index value=%d > 99.	The maximum malfunction index value is 99. Use the procedure in Appendix B.5 to modify the malfunction index value.[4]
TGM: #27 Err. Event=%s is Unknown. Read manual.	It is almost impossible for the driver to distinguish between ‘Return to Normal’ events and those events not associated with a malfunction. When a non-malfunction event is found that doesn’t match the entries in the Return to Normal Table then this message is printed. It is printed 20 times and then suppressed. Ignore the messages if they clearly are not ‘Return to Normal’ events that correspond to Malfunctions. If, however, they are obviously associated with Malfunctions then add the description to the ‘Return to Normal’ Table. Notes in Appendix B.6 provide information on how to do this.
TGM: #28 Err TGM_Function=%s invalid.	This message is printed during Map Descriptor Validation. The function is not recognized. Correct the configuration using information found in section 4.4.2[4]

[4] Modify the configuration CSV file. Download the modified file to the FieldServer and Reset the FieldServer for the changes to take effect.

Appendix C.1. Exposing Driver Operating Statistics

In addition to the standard FieldServer operating statistics the driver exposes certain key stats in a Data Array if required. These stats can then be monitored by an upstream device.

Add the following to your configuration file to activate these stats

```
Expose Driver Operating Stats.

Data_Arrays
Data_Array_Name,               Data_Format,        Data_Array_Length
ATMI-TGM-stats,                UINT16,             1000
```

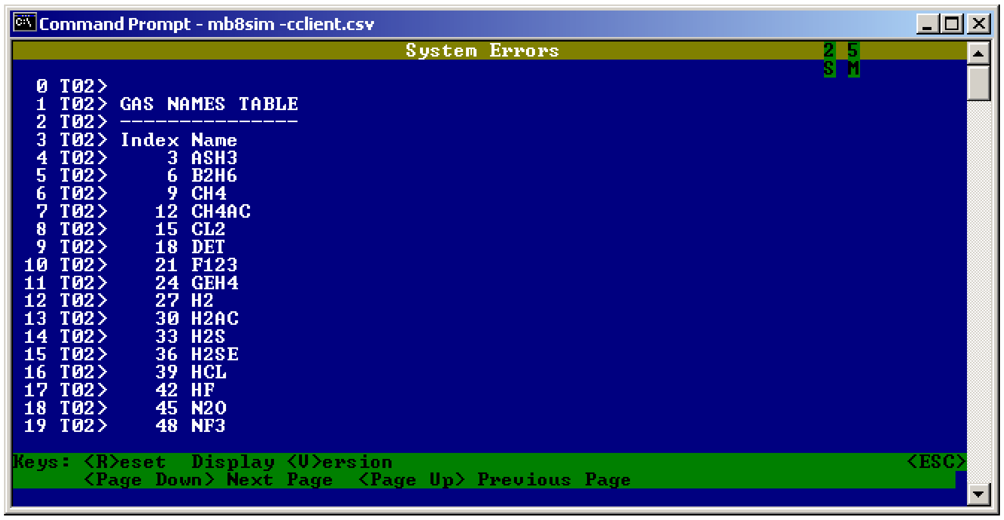
To calculate the actual offset multiply the port number by 100 and add the relative offset. Items marked * do not consider the port number in calculating the actual offset.

Stat Relative Offset	Notes
1	Increments each time client sends a poll.
2	Total of all the bytes sent to the TGM device
3	Increments each time the client receives a complete response as indicated by the prompt character
4	Increments each time the client receives a CR-lf pair in a response
5	Count of all the bytes received by the client in response to TGM polling
6	Count of all the messages that were sent to the parse function.
8	Count of all the message that were parsed successfully
9	Increments by 1 each time a poll times out
10	*When set to 1 the driver dumps the contents of its gas name lookup table to the error log. Driver resets to zero after dump.
11	*When set to 1 the driver dumps the contents of its gas engineering units lookup table to the error log. Driver resets to zero after dump.
12	*When set to 1 the driver dumps the contents of its CALR gas lookup table to the error log. Driver resets to zero after dump.
13	*When set to 1 the driver dumps the contents of its Malfunction Lookup table to the error log. Driver resets to zero after dump.
14	*When set to 1 the driver dumps the contents of its 'Return to Normal' Lookup table to the error log. Driver resets to zero after dump.
15	When set to 1 the driver does not use the soundex function.
16	Increments by 1 each time a response line is too large to fit the connection in buffer
17	Increments by 1 each time a response is too large to fit the in the report buffer

To execute a table dump do the following

- Connect to the FieldServer using RUINET
- Browse to the Array called “ATMI-TGM-stats”
- Modify the value found at offset by typing M x 1 <Enter>
  Where x = 10 or 11 or 12 or 13 or 14

A typical dump is shown below



```
Command Prompt - mb8sim -cclient.csv
                                  System Errors                          2 5
                                                                         S M
  0 T02>
  1 T02> GAS NAMES TABLE
  2 T02> ---------------
  3 T02> Index Name
  4 T02>     3 ASH3
  5 T02>     6 B2H6
  6 T02>     9 CH4
  7 T02>    12 CH4AC
  8 T02>    15 CL2
  9 T02>    18 DET
 10 T02>    21 F123
 11 T02>    24 GEH4
 12 T02>    27 H2
 13 T02>    30 H2AC
 14 T02>    33 H2S
 15 T02>    36 H2SE
 16 T02>    39 HCL
 17 T02>    42 HF
 18 T02>    45 N2O
 19 T02>    48 NF3
Keys: <R>eset  Display <V>ersion                                          <ESC>
      <Page Down> Next Page  <Page Up> Previous Page
```

Appendix D. Troubleshooting tips

Appendix D.1. Connection Tips & Hints

The ATMI-TGM's serial ports are configured using the TGM's configuration file – typically named config.dat

Search for the following section. The port you wish to connect to must be set as a "Remote Console".

```
/* the channel utilization array specifies how a parallel (or serial) */
/* channel is to be utilized.  The possible entries are:              */
/*    0 = this channel is forced out of service (not implemented, etc)*/
/*    1 = this channel is a printer device                            */
/*    2 = this channel is a remote console                            */
/* There can be more than one printer device (max = all 3 channels), */
/* but there can be only one remote console (COM1: or COM2 :).        */
/* Leave the Acoustic Sensor entry = 0 (controlled by bvCous)         */  /* PMC */
BYTE    chanUtil[] =    {
            0,        // [0] = local (parallel) printer
            0,        // [1] = COM1: serial channel
            2,        // [2] = COM2: serial channel
            0,        // [3] = Acoustic Sensor serial channel
            };

/* the baudRate table specifies the baud rate for the serial channels */
/* can be one of: 110, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400 */
/* an entry of 0 in chanUtil[] overrides the entry here               */ /* PMC, too slow and wont finish printing in time
*/
WORD    baudRate[] =    {
            XX,      // [0]
            9600,    // [1]     // COM1:
            9600,    // [2]     // COM2:
            9600,    // [3]     // cous chan
            };
```

THIS PAGE INTENTIONALLY LEFT BLANK